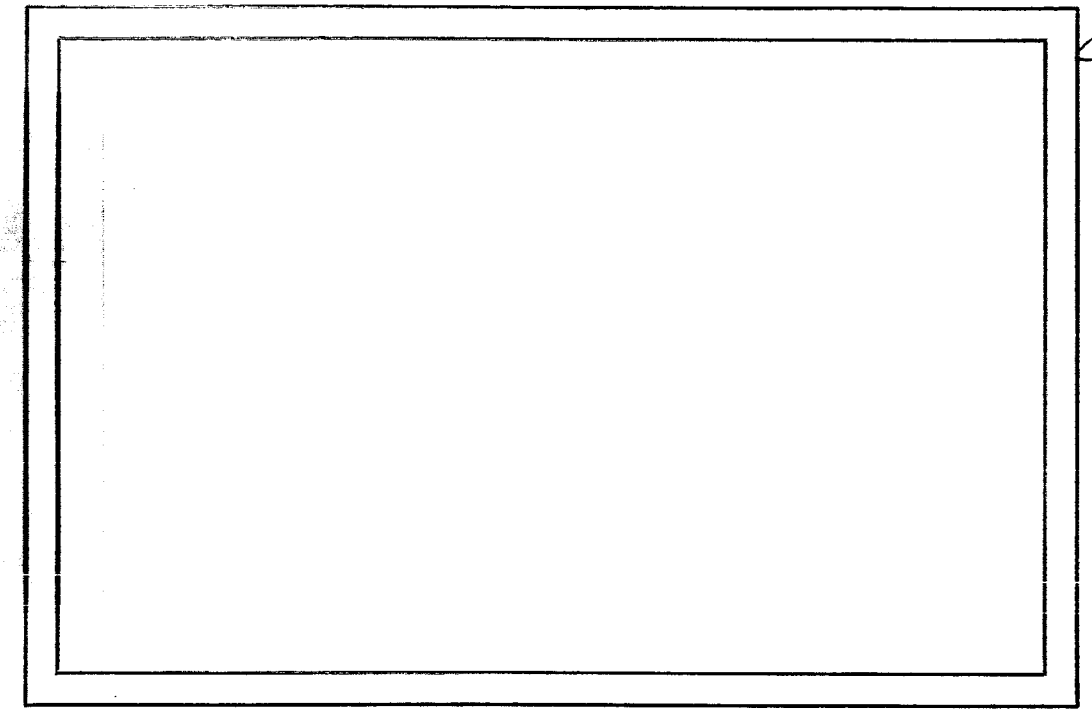
N64-16753
CODE-1
CR-53080



OTS PRICE
XEROX $ 4.60 ph
MICROFILM $ 1.64 mf.

UNPUBLISHED PRELIMINARY DATA

1485994

UNIVERSITY OF MARYLAND

COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

(NASA CR-53080; TR-64-5) OTS: $4.60 ph, $1.64 mf

[X] OTS

[2]

(NASA Grant NsG-398) Technical Report TR-64-5

January 1964 48p

MOIST*:

Macro Output Input SysTem for the IBM 7090

Typist: Keep the underlining

by

Gerald M. Berns (IBM)
IBM Corporation

*The computer time for this project was supported by the National Aeronautics and Space Administration grant NsG-398. The author performed the work while assigned to the Computer Science Center as the IBM Systems Representative.

ABSTRACT

16753

This report describes the Macro Output-Input System, a programming package for the IBM 7090 designed to make the tasks of reading input and writing output as easy to program as simple addition and subtraction. The advantages and capabilities of the system are described along with numerous examples illustrating its usage.

AUTHOR

MOIST

The Macro Output-Input SysTem

Purpose of the System:

The Macro Output-Input SysTem ("MOIST"), was designed to make the tasks of reading input and writing output as easy for the programmer to code as addition and subtraction. The concept is to free the 7090 programmer entirely from such chores as having to code his own individually tailored machine language input-output routines, having to improvise page eject and pagination routines, or having to write Fortran subroutines for data transmission, thereby leaving him more time to concentrate on other facets of his work. Using MOIST, input and output is accomplished conveniently by macro instructions which are coded by the programmer in the same manner as he codes the ordinary machine instructions. In order to maintain system simplicity, the number of MOIST instructions necessary to perform each data transmission has been kept to a minimum; in most cases only one MOIST instruction is required to read or write a desired amount of data in the desired form. The total number of different macro instructions that comprise MOIST was purposely kept small so as to avoid confusing the programmer with an abundance of new and strange instructions. In order to use MOIST efficiently, the programmer need not be familiar with the actual construction and inner operation of the macro instructions themselves.

A programmer, like most other people, can usually be counted on to greet any new thing with reluctance, and to find a large number of reasons why the new thing should not be used. For this reason it was recognized that the major factor affecting early programmer acceptance of the Macro Output-Input System is not that it must be simple to use, (this, too, of course), but that the programmer must be convinced that it is easy (and desirable) to use. For this reason the MOIST Manual (which completely explains how to use MOIST) practices what it preaches: the MOIST Manual is itself the output from a 7090 program which was written in IBMAP and which uses MOIST exclusively for all its data transmissions.

Description and Advantages of the System:

MOIST enables the programmer to accomplish data input and output easily and efficiently without leaving either the IBMAP or the FAP assembly language. Twelve macro instructions (two others are nested macros used internally by MOIST) are embodied in the system. These instructions have the capability to read and write numeric, alphanumeric, and tabular data. A page eject and page numbering routine is built into the system, and MOIST handles these chores (allowing 59 lines of output from MOIST per page) automatically with no programmer intervention whatsoever. All output can be automatically centered across the page, left- or right-justified on the page, or placed in any other position on the page--all at the programmer's option.

Alphanumeric output of up to 114 characters can be accomplished using the single instruction, PRINT, followed by an ETC card. A single PRINT statement with no following ETC card can handle up to 57 characters of output. Output of the form "BETA= 22.3243" can also be accomplished on one coded line using the instruction WRITE or WRITEC. Coding the operation HEAD (HEDD in MOIST for FAP, called MOIST-F), followed by an alphanumeric message of up to 60 characters, produces a heading, consisting of the message, which will appear at the top of every following output page, automatically centered across the page. This also requires but one line of code. HEAD (or HEDD) may be coded in a program as many times as it is desired to change the heading. Lines may be skipped on the output page using SKIP, and a page may be ejected at the programmer's option (thereby overriding the automatic pagination routine) by coding NUPAGE. Output of numeric and alphanumeric data in columnar format may be handled by using the DATA instruction. DATA will output from 1 to 10 columns of data centered across the page, and will divide the columns of data, which may be of any length, into any desired block size (where each block of data is separated from the next by two blank lines) at the programmer's option. DATA also has built into it an error routine, incorporating a diagnostic message, which takes action when certain programmer or keypunch errors occur. If the programmer desires to have column headings centered above the columns of data, which are themselves centered on the page, he may use the instruction COLHED. COLHED accepts column headings of up to 11 alphanumeric characters.

In addition, MOIST offers the programmer an instruction which is especially designed for debugging (WRITE, WRITEC, and DATA all have debugging applications as well). SNAP, when coded in any one of its three forms, saves and produces on the output page the contents of the accumulator (P and Q bits included), the multiplier-quotient, the sense indicators, index registers 1, 2, and 4 (MOIST has not yet been adapted for the 7094), and the location counter.

The system is so constructed that each of the 12 macro instructions available to the programmer organizes and stores the information contained in its arguments and then transfers for verification and diagnostic action (if any) and data transmission to one or more of four subroutines. These subroutines are "Most", which implements the HEAD (HEDD in MOIST-F), PRINT, PRINTC, SKIP, INDENT, NUPAGE, DATA, and COLHED instructions and which requires 883 core storage locations (890 in MOIST-F); "Writ", which implements the WRITE and WRITEC instructions and requires 101 core locations (in both systems); "Inpt", which implements the INPUT instruction and requires 105 core locations (109 in MOIST-F); and "Snap", which implements the SNAP instruction and requires 53 core locations (55 in MOIST-F). All of these subroutines refer to the Fortran IV (Fortran II for MOIST-F) read-write library subroutine which is ultimately responsible for the actual transmission of data. The "Most" subroutine, which also incorporates the automatic page eject and page numbering routines, must be in core at execution time if any output is to be written; "Writ" and "Snap" are only required if the instructions which they implement are present in the user's program. If MOIST is only to be used to read input, the "Inpt" subroutine is all that is required.

One of the major advantages of MOIST is that it is easier to use than Fortran for accomplishing the forms of data transmission that have been described. This point is made particularly clear when the amount of Fortran code that must be written to accomplish data transmission equivalent to some of the MOIST instructions (DATA and COLHED, for example) is examined. Several examples of this are presented in the MOIST Manual, which is included in this report as an appendix. However, it must be made clear that Fortran, being a complete language, is much more flexible than MOIST, which is only an input/output system designed to be used <u>with</u> a language (IBMAP or FAP). In fact, one application of MOIST is its use in an IBMAP subroutine to handle large amounts of I/O for a Fortran IV (or Cobol) main program, or in a FAP subroutine to handle large amounts of I/O for a Fortran II main program.

The MOIST Manual--A Prefatory Note:

The MOIST Manual is actually the output from a 7090 program written in IBMAP, the IBJOB Processor assembly language. The program producing the MOIST Manual uses MOIST exclusively for writing the output. The Manual program occupies 22,500 cells (9,586 decimal) and requires 1.31 min. to assemble using 729 V tape drives. The loader requires 0.65 min. to load the Manual program and the four MOIST subroutines. The execution of the Manual program, which produces the 1,347 line, 41 page MOIST Manual, is accomplished in 0.31 min.

This Manual is unlike other manuals in two respects. First, it is itself a demonstration of the system that it describes. And second, the examples given to illustrate each output instruction are "living" in the sense that they have not been copied from other listings and included here; rather, these are the "other" listings. For this reason it is felt that this Manual can be of more than ordinary assistance to the programmer: not only is every line of the Manual in some sense an example of the system, but, in particular, there are over 30 examples in the Manual, which were chosen expressly to test the system, and to answer many of the typical questions of the kind, "what would happen if this were coded". This has been done because it is felt that this is what a programmer wants to see when he is first presented with a system of this nature. Moreover it was felt that a system which can survive a test of this sort is more likely than not to win the confidence of the programmer and be used by him, and this is, after all, the ultimate aim here.

Also included in the Manual is a fairly extensive comparison of MOIST with FORTRAN. This is in the form of coding examples showing the comparative levels of difficulty in using each system and the advantages obtained by using MOIST.

An attempt has been made to include in the Manual, in words and "pictures", sufficient temptation to interest a programmer in the MOIST system, in addition to giving him all the information necessary to use the system.

APPENDIX

THE MOIST MANUAL

THE MOIST MANUAL

WRITTEN BY

GERALD M. BERNS, IBM

AT THE

UNIVERSITY OF MARYLAND

JANUARY, 1964 EDITION

THE MACRO OUTPUT-INPUT SYSTEM

INTRODUCTION TO THE MACRO OUTPUT-INPUT SYSTEM (MOIST)

THE IBMAP ASSEMBLY LANGUAGE AND THE FAP ASSEMBLY LANGUAGE BOTH SHARE THE DRAWBACK OF OTHER ASSEMBLY LANGUAGES FOR BINARY COMPUTERS -- DATA INPUT AND OUTPUT IS AT LEAST AN INCONVENIENCE AND IS MANY TIMES A DOWNRIGHT DIFFICULTY. TO ESCAPE THE COMPLEXITIES OF DATA INPUT AND OUTPUT FROM WITHIN ASSEMBLY LANGUAGES, MANY PROGRAMMERS HAVE RESORTED TO WRITING FORTRAN SUBROUTINES TO ACCOMPLISH DATA TRANSMISSION. OFTEN THIS SOLUTION TO THE DATA INPUT-OUTPUT PROBLEM HAS BEEN PERFECTLY SATISFACTORY - OTHER TIMES IT HAS LEAD TO COMPLICATIONS WHICH APPROXIMATE IN MAGNITUDE THE ORIGINAL PROBLEM.

THE (M)ACRO (O)UTPUT-(I)NPUT (S)YS(T)EM (''MOIST'') IS WRITTEN IN BOTH IBMAP AND FAP (CALLED MOIST-F) FOR USE WITH PROGRAMS WRITTEN IN IBMAP, FORTRAN IV, COBOL, FAP, OR FORTRAN II. MOIST ENABLES DATA INPUT AND OUTPUT TO BE HANDLED SIMPLY ENOUGH THAT THE DATA TRANSMISSION PORTION OF A PROGRAM BECOMES THE EASIEST PART TO WRITE. MOIST IS EASIER TO USE FOR CODING DATA INPUT AND OUTPUT THAN FORTRAN. MOIST IS FAIRLY FLEXIBLE AND OFFERS MANY COMPLETELY AUTOMATIC FEATURES TO THE PROGRAMMER - SUCH AS AUTOMATIC CENTERING OF ALL PRINTED OUTPUT. IT FREES THE PROGRAMMER FROM THE RIGORS OF CHARACTER COUNTING (AS IN FORTRAN) AND SPACING DATA OUTPUT ON THE PAGE. IN ADDITION, MOIST OFFERS THREE DEBUGGING TOOLS TO THE PROGRAMMER.

'HEAD' ('HEDD'), 'NUPAGE', AND THE PAGINATION RCUTINE

A DEMONSTRATION OF MOIST

HEAD (HEDD IN MOIST-F)

NUPAGE

THE AUTOMATIC PAGINATION ROUTINE

TO PLACE A HEADING OF UP TO 60 ALPHANUMERIC CHARACTERS ATOP EACH OUTPUT PAGE, THE PROGRAMMER USING MOIST WRITES (ONCE) ON HIS CODING FORM --

COL	COL	COL
8	12-16	72
HEAD	(ANY DESIRED HEADING)	

FOR EXAMPLE, THE HEADING THAT APPEARS ATOP THIS PAGE WAS ORIGINATED BY - -

HEAD('HEAD' ('HEDD'), 'NUPAGE'...PAGINATION ROUTINE)

NOTE THAT NEITHER CHARACTER COUNTING NOR SPACING IS NECESSARY, THE CENTERING OF THE HEADING BEING AN AUTOMATIC FEATURE OF MOIST. 'HEAD' ITSELF DOES NOT DIRECTLY PRODUCE ANY OUTPUT -- WHENEVER A FRESH OUTPUT PAGE IS BEGUN, THE HEADING IS AUTOMATICALLY PRINTED, LEAVING TWO BLANK LINES BENEATH.

TO EJECT A FRESH OUTPUT PAGE AT ANY POINT IN HIS PROGRAM, THE PROGRAMMER CODES THE INSTRUCTION 'NUPAGE'. FOR EXAMPLE, THE NEXT INSTRUCTION EXECUTED BY THE 7090 IBMAP PROGRAM PRODUCING THIS WRITE-UP WILL BE --

NUPAGE

'HEAD' ('HEDD'), 'NUPAGE', AND THE PAGINATION ROUTINE

NOTE THAT THE EXECUTION OF 'NUPAGE' PLACES THE HEADING ATOP THE FRESH PAGE ALONG WITH THE NEW PAGE NUMBER.

'NUPAGE' IS TO BE USED BY THE PROGRAMMER ONLY WHEN HE WANTS TO EJECT A NEW PAGE AT A SPECIFIC POINT IN HIS PROGRAM. OTHERWISE, MOIST AUTOMATICALLY BEGINS HIS OUTPUT ON A FRESH PAGE FOR HIM, AND THEREAFTER AUTOMATICALLY EJECTS PAGES AS THEY ARE REQUIRED, ALLOWING UP TO 59 LINES OF OUTPUT FROM MOIST PER PAGE - INCLUDING THE 5 LINES PER PAGE REQUIRED BY THE HEADING, PLACING THE PAGE NUMBER AND THE HEADING ATOP EACH NEW PAGE.

NOTES * * * * *

1. THE HEADING MAY BE CHANGED AS MANY TIMES AS DESIRED IN A PROGRAM BY WRITING NEW 'HEAD' STATEMENTS.
2. OTHER USER OUTPUT ROUTINES MAY USE THE AUTOMATIC PAGINATION ROUTINE OF MOIST AS FOLLOWS --
   A. PRIOR TO WRITING OUTPUT, PLACE THE NUMBER OF LINES TO BE PRINTED, INCLUDING BLANK LINES, IN THE ACCUMULATOR AND
   B. TSX ..LINE,4 THE RETURN IS TO 1,4
3. THE NEGATIVE OF THE NUMBER OF LINES REMAINING ON A PAGE IS CONTAINED IN THE ADDRESS OF CELL .LN. .

THE 'PRINT', 'INDENT', 'SKIP', AND 'PRINTC' INSTRUCTIONS

PRINT

INDENT

SKIP

PRINTC

TO OUTPUT ALPHAMERIC INFORMATION USING MOIST, THE PROGRAMMER CODES THE FOLLOWING --

```
PRINT   (ANY MESSAGE)
```

WHATEVER MESSAGE UP TO 57 CHARACTERS LONG THE PROGRAMMER WRITES WITHIN PARENTHESES WILL APPEAR ON HIS OUTPUT, LEFT JUSTIFIED TO THE MARGIN. FOR EXAMPLE, IF ONE CODES - -

```
PRINT   (THIS IS A DEMONSTRATION OF 'PRINT')
```

HIS OUTPUT IS THE FOLLOWING --

```
THIS IS A DEMONSTRATION OF 'PRINT'
```

IF THE PROGRAMMER DESIRES TO OUTPUT A MESSAGE OF GREATER LENGTH THAN 55 CHARACTERS -- BUT LESS THAN 115 CHARACTERS - HE MAY USE AN ETC CARD, AS FOLLOWS - -

```
  COL     COL                                          COL
   8       16                                          72
   PRINT   (THIS IS.....LONGEST LINE (114 CHARACTERS))
   ETC ( THAT IT IS POSSIBLE.......'PRINT' OPERATION.)
```

THIS PRODUCES - -

```
THIS IS AN EXAMPLE OF THE LONGEST LINE (114 CHARACTERS) THAT IT IS POSSIBLE TO OUTPUT USING THE 'PRINT' OPERATION.
```

WHEN THE ETC CARD IS USED WITH 'PRINT' THE OPENING PARENTHESIS OF THE MESSAGE ON THE 'PRINT' CARD MUST BE IN COLUMN 16. THE OPENING PARENTHESIS ON THE ETC CARD, HOWEVER, MAY OCCUR AS EARLY AS COLUMN 12.

THE 'PRINT', 'INDENT', 'SKIP', AND 'PRINTC' INSTRUCTIONS

** PLEASE NOTE ** CURRENTLY THERE IS AN ERROR IN BOTH IBMAP AND FAP THAT CAUSES 'PRINT' WITH AN ETC CARD TO BE HANDLED INCORRECTLY. UNTIL SUCH TIME AS THE PERTINENT MODIFICATIONS TO THE ASSEMBLERS APPEAR, THE USER DESIRING TO AVAIL HIMSELF OF THIS FEATURE OF MOIST MUST SUBSTITUTE THE INSTRUCTION 'DOUBPR' FOR 'PRINT'. WHEN THE MODIFICATIONS APPEAR THE MOIST MACROS WILL BE CHANGED TO ACCOMODATE THEM.

IF THE PROGRAMMER DESIRES HIS MESSAGE TO BE OTHER THAN LEFT-JUSTIFIED ON THE OUTPUT PAGE, HE MAY USE THE 'INDENT' INSTRUCTION. FOR EXAMPLE, IF HE WANTS A LINE OF OUTPUT TO BE INDENTED 60 SPACES FROM THE LEFT MARGIN, HE CODES

```
INDENT  60
PRINT   (THIS IS INDENTED 60 SPACES)
```

THE FOLLOWING OUTPUT WAS CODED SIMILARLY TO THE ABOVE EXAMPLE --

THIS IS NOT INDENTED

THIS IS INDENTED 10 SPACES

THIS IS INDENTED 30 SPACES

THIS IS INDENTED 60 SPACES

THIS IS INDENTED 90 SPACES

THIS IS INDENTED 103 SPACES

THIS IS INDENTED 104 SPACES

THIS IS INDENTED 105 SPACES

THE 'PRINT', 'INDENT', 'SKIP', AND 'PRINTC' INSTRUCTIONS

THIS IS INDENTED 150 SPACES

THIS IS ALSC INDENTED 150 SPACES**********X

SC IS THIS INDENTED 150 SPACES**********YX**********X

NOTE THAT MOIST WILL NOT 'INDENT' BEYOND THE RIGHT MARGIN REGARDLESS OF THE NUMBER CF SPACES SPECIFIED. WRITING 'INDENT' WITH A NULL VARIABLE FIELD WILL AUTOMATICALLY RIGHT JUSTIFY THE MESSAGE ON THE PRINTED PAGE.

THE 'PRINT' INSTRUCTION ALWAYS SKIPS A LINE BEFORE PRINTING. IF THE PROGRAMMER WISHES TC SKIP MORE LINES BEFORE PRINTING, HE MAY USE THE 'SKIP' INSTRUCTION. FOR EXAMPLE,

```
SKIP    5
INDENT  10
PRINT   (INDENTED 10 AND SKIPPED 5)
```

RESULTS IN

INDENTED 10 AND SKIPPED 5

BY USING THE 'INDENT' AND 'SKIP' INSTRUCTIONS, IN CONJUNCTION WITH 'PRINT' AND OTHER INSTRUCTIONS TO BE DISCUSSED, THE PROGRAMMER HAS COMPLETE CONTROL OVER THE POSITIONING ON THE PAGE OF HIS ALPHAMERIC OUTPUT.

A 'SKIP' OF N LINES ACTUALLY ONLY SKIPS N-1 LINES - THE OTHER SKIPPED LINE(S) IS PRODUCED BY THE NEXT OUTPUT INSTRUCTION.

IF THE PROGRAMMER WISHES ALPHAMERIC OUTPUT TO BE CENTERED ON THE LINE, INSTEAD OF CODING 'INDENT' AND

THE 'PRINT', 'INDENT', 'SKIP', AND 'PRINTC' INSTRUCTIONS

'PRINT' SUCCESSIVELY, HE CODES THE ONE INSTRUCTION 'PRINTC'. FOR EXAMPLE,

```
PRINTC .(•X•)
```

PRODUCES

•X•

'PRINTC' IS USED EXACTLY AS IS 'PRINT', EXCEPT THAT 57 CHARACTERS IS THE MAXIMUM MESSAGE SIZE ALLOWED. 'PRINTC' DISABLES ANY PRECEDING 'INDENT' INSTRUCTION.

NOTES • • • • •

1. WHEN PARENTHESES ARE PART OF THE 'PRINT' OR 'PRINTC' MESSAGE, THEY MUST APPEAR IN PAIRS WITH THE OPENING PARENTHESIS PRECEDING THE CLOSING PARENTHESIS ON EACH CARD THAT CONTAINS THEM.
2. WHEN 'INDENT' RIGHT JUSTIFIES OUTPUT, IT IS ON THE BASIS OF 131 CHARACTERS TO THE LINE.
3. WHEN 'SKIP' AND/OR 'INDENT' ARE USED IT IS NOT REQUIRED THAT THEY IMMEDIATELY PRECEDE THE NEXT OUTPUT INSTRUCTION AS LONG AS THEY ARE CODED PRIOR TO IT.
4. THE CELL SPECIFYING THE AMOUNT OF THE INDENT IS ALWAYS RESET TO ZERO AFTER EACH EXECUTION OF 'INDENT'.
5. WHEN 'SKIP' AND 'INDENT' ARE USED TOGETHER - AS IN THE EXAMPLE - THE 'SKIP' INSTRUCTION MUST ALWAYS PRECEDE THE 'INDENT' INSTRUCTION.
6. A 'SKIP' OF LESS THAN 3 IS CONSIDERED TO BE A 'SKIP' OF 3, AND A 'SKIP' OF GREATER THAN 53 IS

CONSIDERED TO BE A 'SKIP' OF 53.

7. A 'SKIP' OF MORE LINES THAN REMAIN ON THE OUTPUT PAGE RESULTS IN A NEW PAGE BEING EJECTED -REPLETE WITH PAGE NUMBER AND CENTERED HEADING- AND THE FULL 'SKIP' BEING MADE ON THE FRESH SHEET.

THE 'WRITE' AND 'WRITEC' INSTRUCTIONS OF MOIST

WRITE

WRITEC

MANY TIMES THE PROGRAMMER IS REQUIRED TO OUTPUT THE NAME OF A VARIABLE FOLLOWED BY AN EQUALS SIGN FOLLOWED BY THE VALUE OF THE VARIABLE (FOR EXAMPLE, X22 = 26.6000). THE MOIST INSTRUCTION 'WRITE' ENABLES THIS TO BE DONE EASILY.

SUPPOSE IN A PROGRAM THERE ARE 3 PARAMETERS X, Y, Z WHICH HAVE THE VALUES .5, .6, AND .7. THE PROGRAMMER WISHES TO OUTPUT X, Y, AND Z AS DESCRIBED ABOVE, IN E15.8 CONVERSION FORM. USING MOIST HE CODES --

```
WRITE   (X,Y,Z)E
```

WHICH RESULTS IN

```
X       =   0.50000000E 00
Y       =   0.60000000E 00
Z       =   0.70000000E 00
```

THE 'E' FOLLOWING THE CLOSING PARENTHESIS -THE FORMAT INDICATOR- IN THE EXAMPLE SIGNIFIES THAT THE VALUES OF THE VARIABLES ARE TO BE PRINTED ACCORDING TO E15.8 CONVERSION FORM. IF THE FORMAT INDICATOR IS 'I', THE OUTPUT IS ACCORDING TO I12, IF IT IS 'A', THE OUTPUT IS ACCORDING TO A6, AND IF IT IS 'O' THE OUTPUT IS ACCORDING TO O13. IF THE FORMAT INDICATOR IS 'F' (OR NULL), THE OUTPUT IS ACCORDING TO F12.4 CONVERSION FORM.

WITH THE IBMAP VERSION OF MOIST THE USER ALSO HAS THE OPTION OF SPECIFYING HIS OWN FORMAT. FOR EXAMPLE --

THE 'WRITE' AND 'WRITEC' INSTRUCTIONS OF MOIST

```
INDENT  80
WRITE   (X,Y,Z)F3.1
```

PRODUCES --

```
X       =  0.5
Y       =  0.6
Z       =  0.7
```

IF THE USER CHOOSES TO MAKE USE OF THIS OPTION, HIS FORMAT IS SCANNED FOR ERRORS. IF AN ERRONEOUS CONVERSION FORM IS FOUND, A MESSAGE IS PRINTED AND THE USER RECEIVES HIS RESULT PRINTED IN O12, AS IN THE FOLLOWING EXAMPLE --

```
INDENT  55
WRITE   (X,Y)G3.2
```

FORMAT ERROR AT 20320. OUTPUT DELETED IF WIDTH TOO GREAT, GIVEN IN O12 IF INVALID FORMAT STATED

```
X       =  200400000000
Y       =  200463146315
```

USER FORMATS ARE RESTRICTED TO A MAXIMUM LENGTH OF 6 CHARACTERS.

'WRITE' SKIPS A LINE BEFORE PRINTING EACH VARIABLE. BOTH 'INDENT' AND 'SKIP' MAY BE USED WITH 'WRITE' TO POSITION THE OUTPUT.

THE 'WRITEC' INSTRUCTION WILL AUTOMATICALLY CENTER THIS FORM OF OUTPUT ON THE LINE, SUCH THAT THE EQUALS SIGN FROM THE OUTPUT OF SUCCESSIVE 'WRITEC'S WILL BE LINED UP ONE UNDER THE OTHER, REGARDLESS OF THE DIFFERENT CONVERSION FORMS USED. AN 'INDENT' PRECEDING A 'WRITEC' IS DISABLED BY THE 'WRITEC' INSTRUCTION.

```
WRITEC  MM,I
```

THE 'WRITE' AND 'WRITEC' INSTRUCTIONS OF MOIST

```
WRITEC  X              OR              WRITEC  X,F
WRITEC  X,E
WRITEC  X,O
WRITEC  TT+10,A
```

```
        MM      =              1
        X       =         0.5000
        X       =    0.50000000E 00
        X       =    200400000000
        TT+10   =  N OF M
```

THE USER OF THE IBMAP VERSION OF MOIST MAY SIMILARLY SPECIFY HIS OWN FORMAT FOR 'WRITEC'.

ANOTHER USEFUL OPTION WITH 'WRITE' AND 'WRITEC' AVAILABLE ONLY TO THE USER OF THE IBMAP VERSION OF MOIST AT THIS TIME, DUE TO A FAP ASSEMBLER ERROR, IS ILLUSTRATED IN THE FOLLOWING EXAMPLE --

```
WRITE   (A,...,C,J,...,M,R)E
```

```
A       =  -0.09999999E 02
        =  -0.11000000E 02
C       =  -0.12000000E 02
J       =  -0.50000000E 02
        =  -0.51000000E 02
        =  -0.52000000E 02
M       =  -0.52999999E 02
R       =  -0.58000000E 02
```

TO SPECIFY THAT EVERY CELL BETWEEN TWO GIVEN SYMBOLS - AND INCLUDING THE TWO GIVEN SYMBOLS - BE PRINTED IN THE SPECIFIED FORMAT, 3 DOTS (NO MORE AND NO LESS) ARE WRIT-

TEN AS SHOWN. THE VARIABLE TO THE LEFT OF THE 3 DOTS MUST RESIDE IN A LOWER CORE POSITION THAN THE VARIABLE TO THE RIGHT OF THE 3 DOTS FOR THIS FEATURE TO WORK. OTHERWISE ONLY THE VARIABLES BORDERING THE 3 DOTS WILL BE PRINTED.

'WRITE' AND 'WRITEC' SAVE AND RESTORE THE ENTIRE AC, THE MQ, AND INDEX REGISTERS 1, 2, AND 4. THEREFORE, THEY CAN BE QUITE USEFUL TO THE PROGRAMMER AS DEBUGGING TOOLS WHEN HE NEEDS TO KNOW THE CONTENTS OF CRITICAL CELLS AT DIFFERENT POINTS IN HIS PROGRAM.

NOTES • • • • •

1. RELATIVIZED NOTATION USING THE ASTERISK FOR THE POSITION OF A VARIABLE IS NOT RECOMMENDED. THAT IS, CODING OF THE TYPE WRITE *-8,E IS SPECIFICALLY WARNED AGAINST WITHOUT FIRST STUDYING THE 'WRITE' MACRO DEFINITION.
2. AN ARGUMENT MAY EXCEED 6 CHARACTERS. HOWEVER, ONLY THE FIRST 6 CHARACTERS WILL APPEAR ON THE OUTPUT. FOR EXAMPLE, WRITE SAMPLE-6,F WHERE SAMPLE-6 HAS A VALUE OF 0.6500 PRODUCES SAMPLE = 0.6500 .
3. WHEN THE FORMAT INDICATOR IS 'O' (013), MINUS SIGNS ARE INCORPORATED INTO THE FIRST OCTAL DIGIT - FOR EXAMPLE, WRITE Y,O WHERE Y = -1 RESULTS IN Y = 400000000001 .
4. ETC CARDS MAY BE USED - PROVIDING THAT EACH CARD FROM THE 'WRITE' CARD TO THE NEXT TO LAST ETC CARD HAS ARGUMENTS IN EVERY COLUMN FROM AT LEAST

COLUMN 16 THROUGH TO COLUMN 72. THIS IS MANDATORY EVEN THOUGH IN MANY CASES THIS MAY MEAN SPLITTING A SYMBOLIC NAME BETWEEN CARDS.

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

DATA

COLHED

THE MOST POWERFUL INSTRUCTION IN THE MOIST REPERTOIRE IS THE 'DATA' COMMAND. IT IS CAPABLE OF OUTPUTTING UP TO 10 COLUMNS OF TABULAR INFORMATION IN ANY DESIRED FORTRAN CONVERSION FORM, IN BLOCKS OF ANY DESIRED LENGTH. FOR EXAMPLE, SUPPOSE THAT THERE ARE IN A PROGRAM 3 VARIABLES A, B, AND C WITH 12 STORED VALUES OF EACH. SUPPOSE FURTHER THAT THE PROGRAMMER WISHES TO OUTPUT THESE VALUES ACCORDING TO E15.8 IN COLUMNAR FORMAT, WITH THE COLUMNS SUBDIVIDED INTO BLOCKS OF 5 LINES, AND WITH THE ENTIRE OUTPUT CENTERED ON THE PRINTED PAGE. USING MOIST, CODING

```
DATA    (A,B,C)E15.8,12,5
```

IS ALL THAT IS NECESSARY. USING ARBITRARY VALUES FOR A, B, AND C, THIS RESULTS IN --

```
-0.09999999E 02        -0.11000000E 02        -0.12000000E 02
-0.11000000E 02        -0.12000000E 02        -0.13000000E 02
-0.12000000E 02        -0.13000000E 02        -0.13999999E 02
-0.13000000E 02        -0.13999999E 02        -0.15000000E 02
-0.13999999E 02        -0.15000000E 02        -0.16000000E 02

-0.15000000E 02        -0.16000000E 02        -0.16999999E 02
-0.16000000E 02        -0.16999999E 02        -0.18000000E 02
-0.16999999E 02        -0.18000000E 02        -0.19000000E 02
-0.18000000E 02        -0.19000000E 02        -0.20000000E 02
-0.19000000E 02        -0.20000000E 02        -0.20999999E 02

-0.20000000E 02        -0.20999999E 02        -0.22000000E 02
-0.20999999E 02        -0.22000000E 02        -0.23000000E 02
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

CODING DATA (A,B,C)3E15.8,12,5

OR DATA (A,B,C)(E15.8,E15.8,E15.8)12,5 PRODUCES IDENTICALLY THE SAME RESULT.

IF THE PROGRAMMER WISHES INSTEAD TO PRINT THE VALUES OF B ACCORDING TO F6.3 AND TO HAVE HIS OUTPUT IN BLOCKS OF 3 INSTEAD OF 5, HE CODES

DATA (A,B,C)(E15.8,F6.3,L15.8)12,3

-0.09999999E 02	11.000	-0.12000000E 02
-0.11000000E 02	12.000	-0.13000000E 02
-0.12000000E 02	13.000	-0.13999999E 02
-0.13000000E 02	14.000	-0.15000000E 02
-0.13999999E 02	15.000	-0.16000000E 02
-0.15000000E 02	16.000	-0.16999999E 02
-0.16000000E 02	17.000	-0.18000000E 02
-0.16999999E 02	18.000	-0.19000000E 02
-0.18000000E 02	19.000	-0.20000000E 02
-0.19000000E 02	20.000	-0.20999999E 02
-0.20000000E 02	21.000	-0.22000000E 02
-0.20999999E 02	22.000	-0.23000000E 02

IF THE PROGRAMMER WISHES HIS COLUMNS OF DATA TO BE UNBROKEN (IN ONE BLOCK), THE NUMBER DENOTING BLOCK SIZE MAY BE OMITTED.

DATA (MM)I1,10 OR DATA MM,I1,10,10 PRODUCES

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

1
2
3
4
5
6
7
8
9
0

NOTE THAT 'DATA' LEAVES 2 BLANK LINES BEFORE PRINTING, AND 3 BLANK LINES AFTER PRINTING.

'DATA' IS DESIGNED TO LEAVE 10 BLANKS BETWEEN COLUMNS WHEN THE WIDTH OF THE FIELDS BEING OUTPUT WILL ALLOW IT, AS IN THE PRECEDING EXAMPLES. WHEN THE COMBINED FIELD WIDTH IS SO GREAT THAT IT WILL NOT ALLOW 10 SEPARATING BLANKS, 'DATA' AUTOMATICALLY ACCOMODATES IT, AS IN THE FOLLOWING EXAMPLE --

```
     DATA    (J,K,L,M,N,O,P,Q,R,S)E12.5,25,6
```

```
-0.50000E 02 -0.51000E 02 -0.52000E 02 -0.53000E 02 -0.54000E 02 -0.55000E 02 -0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02
-0.51000E 02 -0.52000E 02 -0.53000E 02 -0.54000E 02 -0.55000E 02 -0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02
-0.52000E 02 -0.53000E 02 -0.54000E 02 -0.55000E 02 -0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02
-0.53000E 02 -0.54000E 02 -0.55000E 02 -0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02
-0.54000E 02 -0.55000E 02 -0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02
-0.55000E 02 -0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02

-0.56000E 02 -0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02
-0.57000E 02 -0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02
-0.58000E 02 -0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02
-0.59000E 02 -0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02
-0.60000E 02 -0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02
-0.61000E 02 -0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

-0.62000E 02 -0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02
-0.63000E 02 -0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02
-0.64000E 02 -0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02
-0.65000E 02 -0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02
-0.66000E 02 -0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02
-0.67000E 02 -0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02

-0.68000E 02 -0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02
-0.69000E 02 -0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02 -0.78000E 02
-0.70000E 02 -0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02 -0.78000E 02 -0.79000E 02
-0.71000E 02 -0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02 -0.78000E 02 -0.79000E 02 -0.80000E 02
-0.72000E 02 -0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02 -0.78000E 02 -0.79000E 02 -0.80000E 02 -0.81000E 02
-0.73000E 02 -0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02 -0.78000E 02 -0.79000E 02 -0.80000E 02 -0.81000E 02 -0.82000E 02

-0.74000E 02 -0.75000E 02 -0.76000E 02 -0.77000E 02 -0.78000E 02 -0.79000E 02 -0.80000E 02 -0.81000E 02 -0.82000E 02 -0.83000E 02

HERE THE SEPARATION HAS BEEN DECREASED TO 1 CHARACTER WIDTH (THE MINIMUM ALLOWABLE BY MOIST) IN ORDER TO HANDLE THE OUTPUT. IF THE FIELD WIDTHS PLUS THE REQUISITE NUMBER OF MINIMUM COLUMN SEPARATORS EXCEED 131, AN ERROR MESSAGE IS PRINTED, AS FOR THE FOLLOWING EXAMPLE.

```
DATA    (J,K,L,M,N,O,P,Q,R,S)E13.6,25,6
```

FORMAT ERROR AT 23173. OUTPUT DELETED IF WIDTH TOO GREAT, GIVEN IN 012 IF INVALID FORMAT STATED

IF AN INCORRECT CONVERSION FORM SHOULD BE KEYPUNCHED (ONE WHICH IS NOT E, F, I, O, OR A), MOIST GIVES AN ERROR MESSAGE AND PRINTS THE PARTICULAR PARAMETER(S) ACCORDING TO 012.

```
DATA    (A,B,C)(F8.4,Z11.9)5,4
```

FORMAT ERROR AT 23317. OUTPUT DELETED IF WIDTH TOO GREAT, GIVEN IN 012 IF INVALID FORMAT STATED

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

-10.0000	604540000000	604600C00000
-11.0000	604600000C00	604640C00000
-12.0000	604640C00C00	604700C00000
-13.0000	604700000000	604740C00000
-14.0000	604740000C00	605400C0C000

IN THE EVENT THAT N ARGUMENTS AND M CONVERSION FORMS ARE GIVEN, WHERE M IS GREATER THAN N, THE FIRST N CONVERSION FORMS ARE UTILIZED AND NO ERROR MESSAGE IS GIVEN.

```
      DATA    (J,K,L,M,N,O,P,Q,R)(3F8.4,4E12.5,5F8.4)5,4
```

-50.0000	-51.0000	-52.0000	-0.53000E 02	-0.54000E 02	-0.55000E 02	-0.56000E 02	-57.0000	-58.0000
-51.0000	-52.0000	-53.0000	-0.54000E 02	-0.55000E 02	-0.56000E 02	-0.57000E 02	-58.0000	-59.0000
-52.0000	-53.0000	-54.0000	-0.55000E 02	-0.56000E 02	-0.57000E 02	-0.58000E 02	-59.0000	-60.0000
-53.0000	-54.0000	-55.0000	-0.56000E 02	-0.57000E 02	-0.5800CE 02	-0.59000E 02	-60.0000	-61.0000
-54.0000	-55.0000	-56.0000	-0.57000E 02	-0.58000E 02	-0.59000E 02	-0.60000E 02	-61.0000	-62.0000

THE MOIST INSTRUCTION 'COLHED' GIVES THE PROGRAMMER THE CAPABILITY OF CENTERING COLUMN HEADINGS OVER SPECIFIC COLUMNS OF TABULAR INFORMATION OUTPUT FROM 'DATA', WHICH ARE THEMSELVES CENTERED. 'COLHED' SKIPS ONE LINE BEFORE PRINTING.

```
      COLHED  (TIME,ALTITUDE,TEMPERATURE)
      DATA    (A,B,C)(E15.8,E12.5)10,5
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

TIME	ALTITUDE	TEMPERATURE
-0.09999999E 02	-0.11000E 02	-0.12000E 02
-0.11000000E 02	-0.12000E 02	-0.13000E 02
-0.12000000E 02	-0.13000E 02	-0.14000E 02
-0.13000000E 02	-0.14000E 02	-0.15000E 02
-0.13999999E 02	-0.15000E 02	-0.16000E 02
-0.15000000E 02	-0.16000E 02	-0.17000E 02
-0.16000000E 02	-0.17000E 02	-0.18000E 02
-0.16999999E 02	-0.18000E 02	-0.19000E 02
-0.18000000E 02	-0.19000E 02	-0.20000E 02
-0.19000000E 02	-0.20000E 02	-0.21000E 02

BLANK (OR NULL) COLUMN HEADINGS ARE INDICATED AS FOLLOWS --

```
COLHED  (,A,,B,,C)
DATA    (L,M,N,S,Q,R,J)(E15.8,E12.5),10,5
```

A		B		C		
-0.52000000E 02	-0.53000E 02	-0.54000E 02	-0.59000E 02	-0.57000E 02	-0.58000E 02	-0.50000E 02
-0.52999999E 02	-0.54000E 02	-0.55000E 02	-0.60000E 02	-0.58000E 02	-0.59000E 02	-0.51000E 02
-0.54000000E 02	-0.55000E 02	-0.56000E 02	-0.61000E 02	-0.59000E 02	-0.60000E 02	-0.52000E 02
-0.55000000E 02	-0.56000E 02	-0.57000E 02	-0.62000E 02	-0.60000E 02	-0.61000E 02	-0.53000E 02
-0.55999999E 02	-0.57000E 02	-0.58000E 02	-0.63000E 02	-0.61000E 02	-0.62000E 02	-0.54000E 02
-0.57000000E 02	-0.58000E 02	-0.59000E 02	-0.64000E 02	-0.62000E 02	-0.63000E 02	-0.55000E 02
-0.58000000E 02	-0.59000E 02	-0.60000E 02	-0.65000E 02	-0.63000E 02	-0.64000E 02	-0.56000E 02
-0.59000000E 02	-0.60000E 02	-0.61000E 02	-0.66000E 02	-0.64000E 02	-0.65000E 02	-0.57000E 02
-0.59999999E 02	-0.61000E 02	-0.62000E 02	-0.67000E 02	-0.65000E 02	-0.66000E 02	-0.58000E 02
-0.61000000E 02	-0.62000E 02	-0.63000E 02	-0.68000E 02	-0.66000E 02	-0.67000E 02	-0.59000E 02

'COLHED' WILL CENTER COLUMN HEADINGS WHENEVER SPACE EXISTS TO DO SO. FOR EXAMPLE,

```
COLHED(AAAAAAAAAAA,BBBBBBBBBBB,.....,EEEEEEEEEEE,X)
DATA    (MM,MM,MM,MM,MM,MM)11,10,3
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

```
AAAAAAAAAAA BBBBBBBBBBB CCCCCCCCCCC DDDDDDDDDDD EEEEEEEEEEE X

1          1          1          1          1          1
2          2          2          2          2          2
3          3          3          3          3          3

4          4          4          4          4          4
5          5          5          5          5          5
6          6          6          6          6          6

7          7          7          7          7          7
8          8          8          8          8          8
9          9          9          9          9          9

0          0          0          0          0          0
```

'COLHED' WILL CENTER COLUMN HEADINGS OVER THE INDICATED COLUMNS EVEN IF SOME OVERLAP ABOVE ADJACENT COLUMNS IS NECESSARY.

```
     COLHED  (,ABCDEFGHIJK,,LMNOPQRSTUV,,,WXYZ1234567,, )
(              890AAABBBCC)
     DATA    (J,K,L,M,N,O,P,Q,R,S)(E15.8,F4.0,E15.8,     )
(              F4.0,2E15.8,F4.0,E15.8,F4.0,E15.8)20,5
```

ABCDEFGHIJK		LMNOPQRSTUV				WXYZ1234567		890AAABBBCC	
-0.50000000E 02	-51.	-0.52000000E 02	-53.	-0.54000000E 02	-0.55000000E 02	-56.	-0.57000000E 02	-58.	-0.59000000E 02
-0.51000000E 02	-52.	-0.52999999E 02	-54.	-0.55000000E 02	-0.55999999E 02	-57.	-0.58000000E 02	-59.	-0.59999999E 02
-0.52000000E 02	-53.	-0.54000000E 02	-55.	-0.55999999E 02	-0.57000000E 02	-58.	-0.59000000E 02	-60.	-0.61000000E 02
-0.52999999E 02	-54.	-0.55000000E 02	-56.	-0.57000000E 02	-0.58000000E 02	-59.	-0.59999999E 02	-61.	-0.62000000E 02
-0.54000000E 02	-55.	-0.55999999E 02	-57.	-0.58000000E 02	-0.59000000E 02	-60.	-0.61000000E 02	-62.	-0.63000000E 02
-0.55000000E 02	-56.	-0.57000000E 02	-58.	-0.59000000E 02	-0.59999999E 02	-61.	-0.62000000E 02	-63.	-0.64000000E 02
-0.55999999E 02	-57.	-0.58000000E 02	-59.	-0.59999999E 02	-0.61000000E 02	-62.	-0.63000000E 02	-64.	-0.65000000E 02
-0.57000000E 02	-58.	-0.59000000E 02	-60.	-0.61000000E 02	-0.62000000E 02	-63.	-0.64000000E 02	-65.	-0.66000000E 02
-0.58000000E 02	-59.	-0.59999999E 02	-61.	-0.62000000E 02	-0.63000000E 02	-64.	-0.65000000E 02	-66.	-0.66999999E 02
-0.59000000E 02	-60.	-0.61000000E 02	-62.	-0.63000000E 02	-0.64000000E 02	-65.	-0.66000000E 02	-67.	-0.68000000E 02

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

ABCDEFGHIJK	LMNOPQRSTUV	WXYZ1234567	890AAABBBCC
-0.59999999E 02 -61.	-0.62000000E 02 -63.	-0.64000000E 02	-0.65000000E 02 -66.	-0.66999999E 02 -68.	-0.69000000E 02
-0.61000000E 02 -62.	-0.63000000E 02 -64.	-0.65000000E 02	-0.66000000E 02 -67.	-0.68000000E 02 -69.	-0.70000000E 02
-0.62000000E 02 -63.	-0.64000000E 02 -65.	-0.66000000E 02	-0.66999999E 02 -68.	-0.69000000E 02 -70.	-0.70999999E 02
-0.63000000E 02 -64.	-0.65000000E 02 -66.	-0.66999999E 02	-0.68000000E 02 -69.	-0.70000000E 02 -71.	-0.72000000E 02
-0.64000000E 02 -65.	-0.66000000E 02 -67.	-0.68000000E 02	-0.69000000E 02 -70.	-0.70999999E 02 -72.	-0.73000000E 02
-0.65000000E 02 -66.	-0.66999999E 02 -68.	-0.69000000E 02	-0.70000000E 02 -71.	-0.72000000E 02 -73.	-0.73999999E 02
-0.66000000E 02 -67.	-0.68000000E 02 -69.	-0.70000000E 02	-0.70999999E 02 -72.	-0.73000000E 02 -74.	-0.75000000E 02
-0.66999999E 02 -68.	-0.69000000E 02 -70.	-0.70999999E 02	-0.72000000E 02 -73.	-0.73999999E 02 -75.	-0.76000000E 02
-0.68000000E 02 -69.	-0.70000000E 02 -71.	-0.72000000E 02	-0.73000000E 02 -74.	-0.75000000E 02 -76.	-0.77000000E 02
-0.69000000E 02 -70.	-0.70999999E 02 -72.	-0.73000000E 02	-0.73999999E 02 -75.	-0.76000000E 02 -77.	-0.77999999E 02

NOTE THAT, JUST AS 'DATA' WILL NOT PRINT ON A PAGE UNLESS THERE IS SPACE FOR AT LEAST ONE COMPLETE BLOCK OF OUTPUT, 'COLHED' WILL NOT PRINT ON A PAGE UNLESS THERE IS SPACE FOR ONE COMPLETE BLOCK OF DATA BENEATH THE COLUMN HEADINGS. NOTE ALSO THAT, WHEN THE OUTPUT FROM 'DATA' EXTENDS OVER MORE THAN ONE PAGE, COLUMN HEADINGS FROM A PRECEDING 'COLHED' ARE AUTOMATICALLY PLACED OVER THE OUTPUT ON EACH NEW PAGE.

IF, BY MISTAKE, THE PROGRAMMER SPECIFIES MORE COLUMN HEADINGS IN THE 'COLHED' INSTRUCTION THAN HE SPECIFIES ARGUMENTS OF THE NEXT 'DATA' INSTRUCTION, THE ADDITIONAL COLUMN HEADINGS ARE AUTOMATICALLY REPRESSED. IF A COLUMN HEADING WITH IMBEDDED BLANKS IS DESIRED, IT MUST BE PUT WITHIN PARENTHESES IN THE 'COLHED' INSTRUCTION. ALSO, IF A NULL CONVERSION FORMAT IS INDICATED IN THE 'DATA' INSTRUCTION, THE FORMAT IS ASSUMED TO BE E12.5.

COLHED ((TIME (SEC.))(MACH NO.)DISTANCE)

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

```
DATA    (A,B),,5

      TIME (SEC.)          MACH NO.

    -0.10000E 02        -0.11000E 02
    -0.11000E 02        -0.12000E 02
    -0.12000E 02        -0.13000E 02
    -0.13000E 02        -0.14000E 02
    -0.14000E 02        -0.15000E 02
```

IF THE DATA LENGTH PARAMETER IS OMITTED FROM THE 'DATA' INSTRUCTION, ONLY ONE LINE OF OUTPUT IS GENERATED.

```
COLHED ((TIME (SEC))(VEL (FPS))G/GO(ANGLE (DEG)))
DATA    (J,K,L,M)

TIME (SEC)          VEL (FPS)          G/GO          ANGLE (DEG)

-0.50000E 02        -0.51000E 02       -0.52000E 02  -0.53000E 02
```

'DATA' IS LIMITED TO 10 COLUMNS OF OUTPUT. IF MORE THAN 10 ARGUMENTS ARE GIVEN IN A 'DATA' STATEMENT, ONLY THE FIRST 10 ARE ACKNOWLEDGED, AND NO ERROR MESSAGE IS PRINTED. AN EXAMPLE OF A VERY UNLIKELY TRIPLE CODING ERROR FOLLOWS -13 COLUMN HEADINGS, 12 VARIABLES, AND AN INVALID CONVERSION FORM --

```
COLHED  (AA,BB,CC,DD,EE,FF,GG,HH,II,JJ,KK,LL,MM)
DATA    (J,K,L,M,N,O,P,Q,R,S,A,B)89.3,75,15
FORMAT ERROR AT 25236. OUTPUT DELETED IF WIDTH TOO GREAT, GIVEN IN D12 IF INVALID FORMAT STATED
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

AA	BB	CC	DD	EE	FF	GG	HH	II	JJ
606620000000	606630000000	606640000000	606650000000	606660000000	606670000000	606700000000	606710000000	606720000000	606730000000
606630000000	606640000000	606650000000	606660000000	606670000000	606700000000	606710000000	606720000000	606730000000	606740000000
606640000000	606650000000	606660000000	606670000000	606700000000	606710000000	606720000000	606730000000	606740000000	606750000000
606650000000	606660000000	606670000000	606700000000	606710000000	606720000000	606730000000	606740000000	606750000000	606760000000
606660000000	606670000000	606700000000	606710000000	606720000000	606730000000	606740000000	606750000000	606760000000	606770000000
606670000000	606700000000	606710000000	606720000000	606730000000	606740000000	606750000000	606760000000	606770000000	607400000000
606700000000	606710000000	606720000000	606730000000	606740000000	606750000000	606760000000	606770000000	607400000000	607404000000
606710000000	606720000000	606730000000	606740000000	606750000000	606760000000	606770000000	607400000000	607404000000	607410000000
606720000000	606730000000	606740000000	606750000000	606760000000	606770000000	607400000000	607404000000	607410000000	607414000000
606730000000	606740000000	606750000000	606760000000	606770000000	607400000000	607404000000	607410000000	607414000000	607420000000
606740000000	606750000000	606760000000	606770000000	607400000000	607404000000	607410000000	607414000000	607420000000	607424000000
606750000000	606760000000	606770000000	607400000000	607404000000	607410000000	607414000000	607420000000	607424000000	607430000000
606760000000	606770000000	607400000000	607404000000	607410000000	607414000000	607420000000	607424000000	607430000000	607434000000
606770000000	607400000000	607404000000	607410000000	607414000000	607420000000	607424000000	607430000000	607434000000	607440000000
607400000000	607404000000	607410000000	607414000000	607420000000	607424000000	607430000000	607434000000	607440000000	607444000000

607404000000	607410000000	607414000000	607420000000	607424000000	607430000000	607434000000	607440000000	607444000000	607450000000
607410000000	607414000000	607420000000	607424000000	607430000000	607434000000	607440000000	607444000000	607450000000	607454000000
607414000000	607420000000	607424000000	607430000000	607434000000	607440000000	607444000000	607450000000	607454000000	607460000000
607420000000	607424000000	607430000000	607434000000	607440000000	607444000000	607450000000	607454000000	607460000000	607464000000
607424000000	607430000000	607434000000	607440000000	607444000000	607450000000	607454000000	607460000000	607464000000	607470000000
607430000000	607434000000	607440000000	607444000000	607450000000	607454000000	607460000000	607464000000	607470000000	607474000000
607434000000	607440000000	607444000000	607450000000	607454000000	607460000000	607464000000	607470000000	607474000000	607500000000
607440000000	607444000000	607450000000	607454000000	607460000000	607464000000	607470000000	607474000000	607500000000	607504000000
607444000000	607450000000	607454000000	607460000000	607464000000	607470000000	607474000000	607500000000	607504000000	607510000000
607450000000	607454000000	607460000000	607464000000	607470000000	607474000000	607500000000	607504000000	607510000000	607514000000
607454000000	607460000000	607464000000	607470000000	607474000000	607500000000	607504000000	607510000000	607514000000	056000035721
607460000000	607464000000	607470000000	607474000000	607500000000	607504000000	607510000000	607514000000	056000035721	460000037057
607464000000	607470000000	607474000000	607500000000	607504000000	607510000000	607514000000	056000035721	460000037057	007400437105
607470000000	607474000000	607500000000	607504000000	607510000000	607514000000	056000035721	460000037057	007400437105	100000012346
607474000000	607500000000	607504000000	607510000000	607514000000	056000035721	460000037057	007400437105	100000012346	050730454663

607500000000	607504000000	607510000000	607514000000	056000035721	460000037057	007400437105	100000012346	050730454663	256260546054
607504000000	607510000000	607514000000	056000035721	460000037057	007400437105	100000012346	050730454663	256260546054	605460546054
607510000000	607514000000	056000035721	460000037057	007400437105	100000012346	050730454663	256260546054	605460546054	606060606060
607514000000	056000035721	460000037057	007400437105	100000012346	050730454663	256260546054	605460546054	606060606060	606060606060
056000035721	460000037057	007400437105	100000012346	050730454663	256260546054	605460546054	606060606060	606060606060	606060606060
460000037057	007400437105	100000012346	050730454663	256260546054	605460546054	606060606060	606060606060	606060606060	606060606060
007400437105	100000012346	050730454663	256260546054	605460546054	606060606060	606060606060	606060606060	606060606060	606060606060
100000012346	050730454663	256260546054	605460546054	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
050730454663	256260546054	605460546054	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
256260546054	605460546054	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001
605460546054	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105
606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012365
606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012365	050730264651

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

AA	BB	CC	DD	EE	FF	GG	HH	II	JJ
606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012365	050730264651	635121456031
606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012365	050730264651	635121456031	656060606060
606060606060	002000200001	056000035721	460000037057	007400437105	100000012365	050730264651	635121456031	656060606060	606060606060
002000200001	056000035721	460000037057	007400437105	100000012365	050730264651	635121456031	656060606060	606060606060	606060606060
056000035721	460000037057	007400437105	100000012365	050730264651	635121456031	656060606060	606060606060	606060606060	606060606060
460000037057	007400437105	100000012365	050730264651	635121456031	656060606060	606060606060	606060606060	606060606060	606060606060
007400437105	100000012365	050730264651	635121456031	656060606060	606060606060	606060606060	606060606060	606060606060	606060606060
100000012365	050730264651	635121456031	656060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
050730264651	635121456031	656060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
635121456031	656060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001
656060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105
606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012404
606060606060	606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012404	050730444631

606060606060	606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012404	050730444631	626360606060
606060606060	606060606060	002000200001	056000035721	460000037057	007400437105	100000012404	050730444631	626360606060	606060606060
606060606060	002000200001	056000035721	460000037057	007400437105	100000012404	050730444631	626360606060	606060606060	606060606060
002000200001	056000035721	460000037057	007400437105	100000012404	050730444631	626360606060	606060606060	606060606060	606060606060
056000035721	460000037057	007400437105	100000012404	050730444631	626360606060	606060606060	606060606060	606060606060	606060606060
460000037057	007400437105	100000012404	050730444631	626360606060	606060606060	606060606060	606060606060	606060606060	606060606060
007400437105	100000012404	050730444631	626360606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
100000012404	050730444631	626360606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
050730444631	626360606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060
626360606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	007400436757
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	007400436757	633025604421
606060606060	606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	007400436757	633025604421	235146604664
606060606060	606060606060	606060606060	606060606060	606060606060	002000200001	007400436757	633025604421	235146604664	634764634031
606060606060	606060606060	606060606060	606060606060	002000200001	007400436757	633025604421	235146604664	634764634031	454764636062

'DATA' MAY ALSO BE USED BY THE PROGRAMMER DURING THE DEBUG PHASE OF A PROGRAM FOR SECTIONAL DUMPS IN MORE THAN ONE CONVERSION FORM. FOR EXAMPLE, TO DUMP FROM XX TO XX+19 AND FROM Y TO Y+19 IN OCTAL, DECIMAL, INTEGER, AND BCD FORM, ONE MIGHT WRITE --

```
COLHED((XX IN OCTAL)DECIMAL,INTEGER,BCD(Y IN.)..BCD)
DATA(XX,XX,XX,XX,Y,Y,Y,Y)(012,E15.8,I12,A6,,A6)20,10
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

XX IN OCTAL	DECIMAL	INTEGER	BCD	Y IN OCTAL	DECIMAL	INTEGER	BCD
0C7400437105	0.18849111E-36	1606779973	0I0LZ5	200463146315	0.60000000E 00	17260399821	+4T'T
1CC000025627	0.00004507E-19	8589945751	8002 G	200546314632	0.70000000E 00	17273821594	+5UIC+
050730606060	0.29835470E-26	5492640816	57H	000000000001	0.00000001E-39	1	000001
606031266031	-0.03169445E 02	-17991822361	IF I	000000000002	0.00000003E-39	2	000002
636062304664	-0.10566390E 09	-21219609012	T SHOU	000000000003	0.00000004E-39	3	000003
432460222560	-0.11720692E-30	-3569427824	LD BE	000000000004	0.00000007E-39	4	000004
242562315125	0.12428585E 11	21840370261	DESIRE	000000000005	0.00000009E-39	5	000005
246063302163	0.02758407E 12	22293611635	D THAT	000000000006	0.00000012E-39	6	000006
606330256046	-0.27042498E 02	-18041887782	THE O	000000000007	0.00000015E-39	7	000007
646347646360	-0.12446017E 12	-22340914416	UTPUT	000000000010	0.00000016E-39	8	000008
265146446014	0.18045055E 16	24320297996	FROM '	000000000011	0.00000018E-39	9	000009
242163211460	0.38677770E 10	21773488944	DATA'	000000000012	0.00000019E-39	10	000000
056000035723	0.00002360E-24	6174030803	5 03 C	604500000000	-0.09999999E 02	-17800626176	N0000
460000037057	-0.00009810E-24	-6442466863	0003Y	604540000000	-0.11000000E 02	-17809014784	N-000
0C7400437105	0.18849111E-36	1606779973	0I0LZ5	604600000000	-0.12000000E 02	-17817403392	00000
1CC000025645	0.00004512E-19	8589945765	8002 N	604640000000	-0.13000000E 02	-17825792000	0-000
050730454663	0.29824470E-26	5492595123	57HNOT	604700000000	-0.13999999E 02	-17834180608	P0000
602225602325	-0.11699587E 01	-17487561941	BE CE	604740000000	-0.15000000E 02	-17842569216	P-000
456325512524	-0.08629086E-24	-6230021460	NTERED	605400000000	-0.16000000E 02	-17918066688	*0000
604645606330	-0.13180079E 02	-17827302616	ON TH	605420000000	-0.16999999E 02	-17922260992	*+000

SINCE THE FORTRAN II READ AND WRITE ROUTINES ARE USED BY MOIST-F, IT IS IMPORTANT TO REMEMBER THAT FIXED POINT INTEGERS MUST BE IN THE DECREMENT OF THE WORD WHEN THE I CONVERSION FORM IS USED. SINCE THE INTEGER IS ONLY A HALF-WORD, 17 MIGHT BETTER BE USED IN THE ABOVE EXAMPLE FOR MOIST-F, INSTEAD OF 112.

IF IT SHOULD BE DESIRED THAT THE OUTPUT FROM 'DATA' NOT BE CENTERED ON THE PAGE, THE 'DATA' STATEMENT MAY BE PRECEDED BY AN 'INDENT'.

```
INDENT  10                      COLHED  (A,B,C)
COLHED  (A,B,C)        OR       INDENT  10
DATA    (A,B,C),,10             DATA    (A,B,C),,10
```

THE 'DATA' AND 'COLHED' INSTRUCTIONS OF MOIST

A	B	C
-0.10000E 02	-0.11000E 02	-0.12000E 02
-0.11000E 02	-0.12000E 02	-0.13000E 02
-0.12000E 02	-0.13000E 02	-0.14000E 02
-0.13000E 02	-0.14000E 02	-0.15000E 02
-0.14000E 02	-0.15000E 02	-0.16000E 02
-0.15000E 02	-0.16000E 02	-0.17000E 02
-0.16000E 02	-0.17000E 02	-0.18000E 02
-0.17000E 02	-0.18000E 02	-0.19000E 02
-0.18000E 02	-0.19000E 02	-0.20000E 02
-0.19000E 02	-0.20000E 02	-0.21000E 02

AS SHOWN PREVIOUSLY, MOIST WILL NOT 'INDENT' BEYOND THE RIGHT MARGIN REGARDLESS OF THE NUMBER OF SPACES SPECIFIED.

```
INDENT  150
COLHED  (A,,C)
DATA    (A,B,C),,10
```

A		C
-0.10000E 02	-0.11000E 02	-0.12000E 02
-0.11000E 02	-0.12000E 02	-0.13000E 02
-0.12000E 02	-0.13000E 02	-0.14000E 02
-0.13000E 02	-0.14000E 02	-0.15000E 02
-0.14000E 02	-0.15000E 02	-0.16000E 02
-0.15000E 02	-0.16000E 02	-0.17000E 02
-0.16000E 02	-0.17000E 02	-0.18000E 02
-0.17000E 02	-0.18000E 02	-0.19000E 02
-0.18000E 02	-0.19000E 02	-0.20000E 02
-0.19000E 02	-0.20000E 02	-0.21000E 02

NOTES * * * * *

1. ETC CARDS MAY BE USED WITH BOTH 'DATA' AND 'COLHED'. EVERY CARD BUT THE LAST ETC CARD MUST HAVE ARGUMENTS EXTENDING THROUGH COLUMN 72, EVEN

THOUGH THIS MAY MEAN SPLITTING A SYMBOLIC NAME BETWEEN CARDS.

2. THE RELATIVE POSITION OF ONE VARIABLE WITH RESPECT TO ANOTHER MAY BE USED AS AN ARGUMENT OF THE 'DATA' INSTRUCTION (FOR EXAMPLE, SAM+3). IT IS NOT RECOMMENDED THAT RELATIVIZED NOTATION USING THE ASTERISK (FOR EXAMPLE, *+5) FOR THE POSITION OF A VARIABLE BE USED AS AN ARGUMENT WITHOUT FIRST STUDYING THE 'DATA' MACRO DEFINITION.
3. CONVERSION FORMS SPECIFIED MAY NOT EXCEED 6 CHARACTERS IN LENGTH, NOR MAY ANY CONVERSION FORM BE PREFIXED BY A NUMBER OTHER THAN 1 THROUGH 9.
4. EACH COLUMN HEADING MUST BE FEWER THAN 12 CHARACTERS.
5. IT IS NOT NECESSARY THAT 'COLHED' IMMEDIATELY PRECEDE 'DATA' FOR IT TO BE EFFECTIVE. ANY NUMBER AND ALL TYPES OF INSTRUCTIONS -EXCEPT ANOTHER 'COLHED'- MAY SEPARATE THEM.
6. TO INSURE REGULAR SPACING OF OUTPUT AND CENTERED COLUMN HEADINGS, THE PROGRAMMER SHOULD NOT INCLUDE SPACING CONSIDERATIONS IN THE DETERMINATION OF THE WIDTH PORTION OF THE CONVERSION FORM TO BE USED.

THE 'SNAP' INSTRUCTION OF MOIST

SNAP

IF AT ANY POINT IN THE DEBUG PHASE OF A PROGRAM, THE PROGRAMMER WISHES TO KNOW THE ENTIRE CONTENTS OF THE AC, THE MQ, THE SENSE INDICATORS, AND THE THREE INDEX REGISTERS, HE MAY INSERT AT THE DESIRED POINT IN HIS SOURCE DECK A CARD WITH OPERATION CODE 'SNAP' IN COLUMNS 8 THROUGH 11, AS FOLLOWS --

SNAP UR SNAP 0

Q, P BITS	AC (S,1-35)	MQ	INDICATORS	XR1 AND XR2	XR4 AND LOC
0	000000037056	000000000000	000000000060	077251051655	050701027114

ALL OUTPUT FROM THIS FORM OF 'SNAP' IS OCTAL. XR1 OCCUPIES THE DECREMENT, AND XR2 THE ADDRESS, OF THE FIFTH WORD, AND XR4 AND THE LOCATION OF THE SECOND WORD OF THE 'SNAP' MACRO OCCUPY THE SIXTH WORD. WHEN BOTH THE Q AND P BITS OF THE AC ARE ON, THEIR VALUE IS PRINTED AS 3. WHEN THE Q BIT ONLY IS ON, THEIR VALUE IS 2, AND WHEN THE P BIT ONLY IS ON THEIR VALUE IS 1. WHEN BOTH THE Q AND P BITS ARE OFF, THEIR VALUE IS 0.

IF THE PROGRAMMER WANTS THE AC AND MQ PRINTED IN INTEGER (I12) FORM, HE CODES

SNAP I

Q, P BITS	AC (S,1-35)	MQ	INDICATORS	XR1 AND XR2	XR4 AND LOC
0	15918	0	000000000060	077251051655	050444027351

THE 'SNAP' INSTRUCTION OF MOIST

TO GET THE AC AND MQ ACCORDING TO E15.8, HE CODES

SNAP E

Q, P BITS	AC (S,1-35)	MQ	INDICATORS	XR1 AND XR2	XR4 AND LOC
0	0.0CC34851E-39	0.	000000000060	077251051655	050405027410

IN ALL CASES, THE OUTPUT FROM 'SNAP' -OTHER THAN THE AC AND MQ- IS ALWAYS OCTAL.

IT IS SUGGESTED THAT THE PROGRAMMER USING MOIST HAVE 'SNAP' CARDS IN HIS DESK TO INSERT, AS REQUIRED, INTO HIS SOURCE DECK DURING THE DEBUG PHASE. HE MAY DO THIS WITH COMPLETE IMPUNITY PROVIDED THAT HE DOES NOT INSERT A 'SNAP' CARD BETWEEN INSTRUCTIONS WHICH REFERENCE EACH OTHER -- FOR EXAMPLE, BETWEEN TXH *+4,1,1 AND CLA X. THIS WOULD DESTROY THE PROPER FLOW OF THE PROGRAM. HOWEVER, THE NOTATION MAY BE CHANGED TO REFLECT THE FACT THAT 'SNAP' (ALL FORMS) REQUIRES 3 WORDS EACH TIME IT IS CODED.

WHEN 'SNAP I' IS CODED USING MOIST-F, BITS 1 THROUGH 17 ARE CONVERTED TO INTEGER FORM, CONSISTENT WITH THE FAP-FORTRAN II CONVENTION.

THE 'INPUT' INSTRUCTION OF MOIST

INPUT

MOIST HAS ONE INSTRUCTION FOR DATA INPUT, APPROPRIATELY NAMED 'INPUT'. 'INPUT' READS CARD IMAGES (BCD) FROM TAPE, ONLY. TO READ IN ONE VALUE EACH OF J, K, AND L FROM ONE CARD, THE CODE MIGHT BE AS FOLLOWS --

```
INPUT   (J,K,L)3F10.5
```

'INPUT' ESSENTIALLY COMBINES THE READ INPUT AND FORMAT STATEMENTS OF FORTRAN INTO ONE INSTRUCTION, AND ALL FORTRAN RULES RELATING TO CARD FORMATS APPLY STRICTLY TO 'INPUT' AS WELL.

TO READ 25 VALUES EACH (IN CONSECUTIVE ORDER) OF J, K, AND L, ONE CODES

```
INPUT   ((J,,25)(K,,25)(L,,25))3F10.5
```

TO READ 25 SETS OF THE THREE VARIABLES J, K, AND L, ONE CODES

```
INPUT   ((J,1,25),K,(L,1))3F10.5
```

THIS IS EQUIVALENT TO A FORTRAN IMPLIED DO LOOP, AS J(I), K(I), AND L(I) ARE ALL READ FROM THE SAME CARD. TO READ A VALUE FOR M AND THEN M SETS OF J, K, AND L --

```
INPUT   (M(J,1,M)K(L,1))(I10/(3F10.5))
```

THE '1' FOLLOWING THE PARAMETER NAME MUST BE STATED FOR THE FIRST VARIABLE OF THE SET AND FOR THE LAST. AN ALTERNATE WAY OF WRITING THE LAST INSTRUCTION IS

```
INPUT   (M,(J,1,M),K,(L,1))(I10/(3F10.5))
```

NOTES * * * * *

1. ETC CARDS MAY BE USED. THE RULES ARE THE SAME AS

THOSE FOR USING ETC CARDS WITH 'DATA'. HOWEVER, THE 'FORMAT' PORTION OF 'INPUT' MAY NOT EXCEED 58 CHARACTERS.

2. PARENTHESES, WHEREVER USED, MUST BE IN PAIRS.
3. NO MORE THAN 63 DIFFERENT VARIABLE NAMES MAY APPEAR IN ONE 'INPUT' INSTRUCTION.
4. A NUMBER NO GREATER THAN 1000 MAY SPECIFY THE QUANTITY OF PIECES OR SETS OF DATA TO BE READ BY ONE 'INPUT' INSTRUCTION. IF MORE THAN 1000 PIECES OR SETS OF DATA ARE TO BE READ WITH ONE 'INPUT' INSTRUCTION, A SYMBOL MUST BE USED -- SEE THE LAST EXAMPLES.

DESCRIPTION AND EVALUATION OF MOIST

A DESCRIPTION OF MOIST

THE MACRO OUTPUT-INPUT SYSTEM IS TECHNICALLY A GROUP OF 14 MACRO INSTRUCTIONS WHICH STORE INFORMATION AND REFER ACTION TO THE PERTINENT SUBROUTINE. THE ACTUAL DATA TRANSMISSION IS ACCOMPLISHED BY THE FORTRAN LIBRARY READ-WRITE SUBROUTINE. THE MACRO INSTRUCTIONS ARE THE HEART OF THE SYSTEM, MAKING EXTENSIVE USE OF THE 'SET', 'IRP', AND 'IF TRUE' AND 'IF FALSE' PSEUDO-OPERATIONS - ESPECIALLY THE PSEUDO-OP 'SET'. MOIST USES 'SET' IN MANY WAYS BUT PRIMARILY AS A COUNTER DURING THE FIRST ASSEMBLY PASS TO DETERMINE IN ADVANCE THE NUMBER OF CELLS REQUIRED BY THE EXPANSION OF VARIOUS 'IRP'S. THIS IS DONE AS FOLLOWS --

```
A SET     0
  IRP     T
A SET     A+1
  IRP
```

BY RECORDING THE LAST 'SET VALUE' OF 'A' IN A PZE OR TXI INSTRUCTION, THE NUMBER OF CORE LOCATIONS REQUIRED BY THE MACRO DURING THE FINAL ASSEMBLY PASS IS MADE DIRECTLY AVAILABLE TO THE PERTINENT MOIST SUBROUTINE. THIS MAKES POSSIBLE MORE COMPACT CODING IN THE USER'S PROGRAM.

THE MOIST MACROS ARE WRITTEN ON 162 CARD IMAGES (THE MOIST-F MACROS REQUIRE 146). MOIST, APART FROM THE MACROS, IS DIVIDED INTO 4 SEPARATE SUBROUTINES. THE LARGEST SUBROUTINE OF THE SYSTEM, MOST, CONTAINS THE AUTOMATIC PAGING AND PAGE EJECT ROUTINES AND THE IMPLEMENTATION FOR THE 'NUPAGE', 'HEAD' ('HEDD'), 'PRINT',

'PRINTC', 'INDENT', 'SKIP', 'DATA', AND 'COLHED' INSTRUCTIONS. THIS SUBROUTINE REQUIRES 883 CORE LOCATIONS (890 IN MOIST-F). THE SUBROUTINE, WRIT, WHICH IMPLEMENTS 'WRITE' AND 'WRITEC' REQUIRES 101 LOCATIONS (SAME), THE SUBROUTINE WHICH IMPLEMENTS 'SNAP', ITSELF CALLED SNAP, REQUIRES 53 LOCATIONS (55), AND THE SUBROUTINE WHICH IMPLEMENTS 'INPUT', CALLED INPT, REQUIRES 105 CORE LOCATIONS (109). OF THESE SUBROUTINES, MOST AND INPT ARE INDEPENDENT AND MAY EACH BE USED WITHOUT THE OTHER THREE -- WRIT AND SNAP REQUIRE THAT THE SUBROUTINE MOST BE PRESENT. THE ENTIRE MOIST SYSTEM REQUIRES 1142 (DECIMAL) CELLS, AND THE ENTIRE MOIST-F SYSTEM REQUIRES 1155. THE FORTRAN LIBRARY READ-WRITE SUBROUTINE AND THE IOCS SELECTED REQUIRE ADDITIONAL CORE. FURTHER, EACH TIME MOIST INSTRUCTIONS ARE USED IN A PROGRAM THEY REQUIRE CORE, AS FOLLOWS --

MOIST INSTRUCTION	CORE REQUIREMENTS/USE
HEAD	11
NUPAGE	1
PRINT	2 + 10C
INDENT	2 + I
SKIP	1
PRINTC	12
WRITE	5 + 2V
WRITEC	5 + 2V
DATA	3 + V + F
COLHED	2 + 2V

SNAP	3
INPUT	11 + V

WHERE C IS THE NUMBER OF CARDS REQUIRED FOR THE MESSAGE, I IS 1 FOR A UNIQUE 'INDENT' AND ZERO FOR A REPEAT, V IS THE NUMBER OF ARGUMENTS APPEARING IN THE INSTRUCTION, AND F IS THE NUMBER OF FORMATS GIVEN -- IF NO FORMAT IS INDICATED, F = 1.

IN THE INTEREST OF SAVING SPACE, MOIST MAKES EXTENSIVE USE OF THE REMAINING PORTION OF THE DECREMENT OF TSX INSTRUCTIONS IN THE MACRO DEFINITIONS. IN BOTH FAP AND IBMAP THIS USAGE IS GIVEN THE WARNING FLAG, BUT EXECUTION IS NOT AFFECTED.

EVERY MOIST INSTRUCTION SAVES AND RESTORES AT LEAST INDEX REGISTERS 1 AND 2.

ADVANTAGES AND FEATURES OF MOIST

THE ADVANTAGES OF MOIST OVER FORTRAN AND OTHER ASSEMBLY LANGUAGE I/O PACKAGES ARE AS FOLLOWS. MOST IMPORTANTLY, MOIST IS EASIER TO USE AND QUICKER TO CODE. IT IS ALSO VERY CONVENIENT, OUTPUT FROM MOIST BEING ESSENTIALLY PRE-FORMATED, FREEING THE PROGRAMMER FOR MORE IMPORTANT, LESS TIME-CONSUMING WORK. MOIST IS TOLERANT OF CODING OR KEYPUNCH ERRORS - THE PROGRAMMER IS PRACTICALLY ASSURED OF GETTING READABLE OUTPUT REGARDLESS OF CIRCUMSTANCES. IN ADDITION, MOIST CAN BE MORE ECONOMICAL OF CORE THAN FORTRAN FOR JOBS INVOLVING A LARGE NUMBER OF INPUT-OUTPUT STATEMENTS.

THE MOIST MACROS MAY ALSO BE REGARDED AS BUILDING BLOCKS THAT CAN BE COMBINED IN VARIOUS MANNERS TO FORM OTHER MACROS CAPABLE OF PRODUCING SPECIAL FORMS OF OUTPUT.

A COMPARISON OF MOIST WITH FORTRAN IV I/O STATEMENTS

MOIST AND FORTRAN OFFER DIFFERENT INPUT-OUTPUT FEATURES FOR THE USER TO CONSIDER. MOIST IS SIMPLER TO USE, MORE AUTOMATED, AND MORE OUTPUT-PAGE ORIENTED THAN IS FORTRAN, AND FORTRAN IS CONSIDERABLY MORE FLEXIBLE THAN MOIST. THE MAIN ADVANTAGE OF MOIST OVER FORTRAN IS ITS EXTREME EASE AND SPEED OF USE. CONSIDER THE FOLLOWING COMPARISONS MADE WITH FORTRAN IV - THE SAME APPLIES TO FORTRAN II --

FORTRAN IV

```
      WRITE (6,100)(A(I),B(I),C(I),D(I),E(I),I=1,8)
 100 FORMAT (1H //(12XE12.5,12XE12.5,                    ))
     ((  1        12XE12.5,12XE12.5,12XE12.5))
```

MOIST

```
      DATA    (A,B,C,D,E)E12.5,8
```

FORTRAN IV

```
      WRITE (6,101)
```

```
  101 FORMAT (40X,30Hx IS NON-EXISTENT IN THIS CASE//////)

MCIST

          INCENT  40
          PRINT   (X IS NON-EXISTENT IN THIS CASE)
          SKIP    5


FORTRAN IV

          WRITE (6,102)(X(I),Y(I),Z(I),I=1,6)
      102 FORMAT (1H030X4HTIME,30X8HVELOCITY,26X8HDISTANCE///)
     (    1        (24XE15.8,24XF10.5,24XF10.5))

MCIST

          COLHED  (TIME,VELOCITY,DISTANCE)
          DATA    (X,Y,Z)(E15.8,F10.5)6


FORTRAN IV

          WRITE (6,103)
      103 FORMAT (1H036X1HA,43X1HB,43X1HC)
          DO 105 J=1,3
          M=1+(J-1)*6
          N=M+5
      105 WRITE (6,104)(X(I),Y(I),Z(I),I=M,N)
      104 FORMAT (1H0/(29XE15.8,29XE15.8,29XE15.8))
```

```
MOIST
        COLHED  (A,B,C)
        DATA    (X,Y,Z)E15.8,18,6
```

MORE ELABORATE EXAMPLES ARE EASILY CONCEIVED, BUT THESE SUFFICE TO SHOW GRAPHICALLY THE RELATIVE SIMPLICITY, EASE OF USE, AND ORIENTATION TO THE OUTPUT PAGE OF MOIST AS COMPARED WITH FORTRAN. A PROGRAM SUCH AS THE ONE PRODUCING THIS MANUAL, WRITTEN IN IBMAP USING MOIST, WOULD TAKE MANY TIMES AS LONG TO WRITE IN FORTRAN.

DIFFERENCES BETWEEN MOIST AND MOIST-F

THE NEWEST VERSION OF MOIST, MOIST-F (FOR FAP), HAS ALL THE FEATURES OF THE IBMAP VERSION OF MOIST, LESS ONE. THE FEATURE MISSING FROM MOIST-F CONCERNS THE 'WRITE' AND 'WRITEC' INSTRUCTIONS -- WITH MOIST-F THE USER MAY NOT SPECIFY HIS OWN CONVERSION FORM .

THE ONLY TECHNICAL DIFFERENCE IN THE WAY THE TWO SYSTEMS ARE WRITTEN IS NECESSITATED BY THE LIMITATION IN FAP ON THE 'IFF' PSEUDO-OPERATION. 'IFF' IN FAP ONLY MAKES BCD COMPARISONS OF FIELDS, WHILE IBMAP'S 'IFF' AND 'IFT' MAKE BOTH BCD AND S-VALUE COMPARISONS. IN ADDITION, COMPLEX CONDITIONS MAY BE SPECIFIED WITH 'IFT'S AND 'IFF'S IN IBMAP BY JOINING TWO OR MORE OF THEM TOGETHER USING THE LOGICAL OPERATORS 'AND' OR 'OR'. THESE DO NOT EXIST IN FAP. FOR THESE REASONS THE MOIST-F SUBROUTINES

ARE WRITTEN TO HANDLE SOME OF THE TESTING WORK THAT IS ACCOMODATED DIRECTLY IN THE MOIST (IBMAP) MACROS. THUS, TO ACCOMPLISH ONE LESS TASK THE MOIST-F SUBROUTINES REQUIRE SLIGHTLY MORE CORE STORAGE (13 CELLS).

THE 'HEAD' INSTRUCTION OF MOIST IS RENAMED 'HEDD' IN MOIST-F BECAUSE FAP HAS A PSEUDO-OPERATION NAMED 'HEAD' (SUPERCEDED BY 'QUAL' IN IBMAP). FAP ALSO HAS A PSEUDO-OP NAMED 'PRINT' WHICH IS USED WITH UPDATES, BUT IT IS UNLIKELY THAT A CONFLICT OF MEANINGS WILL OCCUR WITH THIS INSTRUCTION.

DESCRIPTION AND EVALUATION OF MOIST

USES OF MOIST

THE DESIGN OBJECTIVE OF MOIST WAS TO SIMPLIFY DATA INPUT AND OUTPUT FOR ALL PROGRAMS TO BE WRITTEN USING IBSYS AND THE IBJOB PROCESSOR OR FAP AND/OR FORTRAN II. THE SPECIFIC AREAS WHERE MOIST CAN MOST ASSIST THE PROGRAMMING EFFORT ARE MANY. SINCE MOIST CAN BE WRITTEN VERY QUICKLY AND EASILY COMPARED WITH FORTRAN, IT WILL HELP IN ALL JOBS WHERE SPEED OF PROGRAMMING IS AN IMPORTANT FACTOR. THE NEATNESS AND COMPOSITION OF OUTPUT FROM MOIST MAKE IT WELL SUITED FOR ALL JOBS WHICH ARE REPORT ORIENTED. ITS EASE OF CODING MAKES MOIST ESPECIALLY USEFUL TO UNIVERSITY STUDENTS AND STUDENTS OF PROGRAMMING. MOIST WILL ALSO AID THOSE JOBS WHICH REQUIRE MANY DIFFERENT DATA INPUT AND OUTPUT STATEMENTS. ANOTHER USEFUL APPLICATION OF MOIST MIGHT WELL BE IN SUBROUTINES TO HANDLE LARGE DATA TRANSMISSIONS FOR FORTRAN MAIN PROGRAMS.

CHANGES TO MOIST SINCE NOVEMBER, 1963 EDITION

1. MOIST FOR FAP (MOIST-F) IS NOW AVAILABLE
2. 59 LINES OF OUTPUT FROM MOIST PER PAGE ARE NOW ALLOWED
3. 'SKIP'S OF 53 LINES/PAGE TO ACCOMODATE 'FOOTERS' ARE ALLOWED
4. WHEN DATA COLUMN AND COLUMN HEADING WIDTHS ARE NOT BOTH EVEN OR ODD, 'COLHED' NOW PLACES THE COLUMN HEADING JUST RIGHT OF CENTER
5. 'INPUT' NOW REQUIRES 11 + V WORDS EACH TIME IT IS CODED, A SAVING OF ONE CELL PER USE
6. 'INDENT 0' NO LONGER RIGHT-JUSTIFIES THE NEXT MESSAGE
7. USER FORMATS ARE NOW ALLOWED WITH 'WRITE' AND 'WRITEC' (IBMAP VERSION ONLY)
8. THE MOIST ERROR MESSAGE HAS BEEN CHANGED